



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/643,331	08/19/2003	Scott S. McDaniel	13940US02	3987
23446	7590	12/07/2010	EXAMINER	
MCANDREWS HELD & MALLEY, LTD			BELANI, KISHIN G	
500 WEST MADISON STREET			ART UNIT	PAPER NUMBER
SUITE 3400			2443	
CHICAGO, IL 60661				
MAIL DATE DELIVERY MODE				
12/07/2010 PAPER				

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 10/643,331

Filing Date: August 19, 2003

Appellant(s): MCDANIEL ET AL.

Philip Henry Sheridan
(Reg.# 59,918)
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 09/27/2010 appealing from the Office action
mailed 04/19/2010.

(1) Real Party in Interest

The examiner has no comment on the statement, or lack of statement, identifying by name the real party in interest in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The following is a list of claims that are rejected and pending in the application:

[1-25]

(4) Status of Amendments After Final

The examiner has no comment on the appellant's statement of the status of amendments after final rejection contained in the brief.

(5) Summary of Claimed Subject Matter

The examiner has no comments on the summary of claimed subject matter contained in the brief.

(6) Grounds of Rejection to be Reviewed on Appeal

The examiner has no comment on the appellant's statement of the grounds of rejection to be reviewed on appeal. Every ground of rejection set forth in the office action from which the appeal is taken (as modified by any advisory actions) is being

maintained by the examiner except for the grounds of rejection (if any) listed under the subheading "WITHDRAWN REJECTIONS."

(7) Claims Appendix

The examiner has no comment on the copy of the appealed claims contained in the appendix to the appellant's brief.

(8) Evidence Relied Upon

US-5,790,804	08-1998	Osborne
US-6,304,910 B1	10-2001	Roach et al.
US-6,421,742 B1	07-2002	Tillier et al.
US-7,376,755 B2	05-2008	Pandya
US-5,991,797	11-1999	Futral et al.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-5, 10, and 16 are rejected under 35 U.S.C. 102(b) as being anticipated by Osborne (U.S. Patent Publication # 5,790,804).

Consider **claim 1**, Osborne shows and discloses a system for transferring data over a remote direct memory access (RDMA) network (Figs. 1 and 2 that show a system with an application 58, at the sender 50 and receiver 52 communicate by sending messages to each other across the network 82; column 7, lines 14-45 disclose the details of the system, including reciting that the operating system may set up a

direct memory access (DMA) with the network interface to extract and copy the message sent from sender's application 1 to message buffer 74 of the receiver's application 1); comprising:

a host comprising a driver and a network interface card (NIC), the driver being coupled to the NIC (Fig. 1 that shows a host (sender 50) with operating system 56, processor (driver) 54, and a network interface 84 that is coupled to the processor; column 7, lines 14-45 disclose the details of the system);

wherein a one-shot initiation process of an RDMA operation is performed between the driver and the NIC of the host (Figs. 1-2 that show the system and list the steps necessary (in Fig. 2) to initiate, by the processor 54 and NIC 84 of host 50, the transfer of a message data from application 1 to the corresponding application 1 at the receiving host 52; column 7, lines 18-36 describe the steps shown in Fig. 2, starting at step 99, by invoking a "Send" command 67 (details shown in Fig. 1) by application 1, which is copied over from the application memory to processor message buffer 62 in step 100, and processing, by processor 54, the message content to a format compatible with the protocol being used in step 102, before handing over the initiation process to NIC 84, that sends the message and its content over the network 82 in step 104, for delivery to a remote receiving host 52; column 7, lines 37-45 further disclose using a direct memory access (DMA) with the network interface to extract and copy the message to message buffer 74 of the remote host 52);

the one-shot initiation process comprising communicating a single command message comprising:

buffer command information (Fig. 1, “Send” command 67 with parameters ID (identifying the receiver of the message), Source Address (location where the message content is stored), and Size (amount of data to be sent) stored in the buffer 66; column 7, lines 19-25 disclose the same details), and

a write command to write a send command (Fig. 2, step 100 that shows a copy (write) command to copy message data from application memory to processor’s message buffer 62 (see Fig. 1)).

Consider **claim 2**, and **as it applies to claim 1 above**, Osborne further shows and discloses the claimed system wherein the driver posts the single command message to perform the one-shot initiation process (Fig. 1, processor (driver) 54 that posts the Send(ID, Source Address, Size) command 67 to Network Interface 84, by copying the command from application 1 memory 66 to message buffers 62-64 of the processor’s operating system 56; Fig. 2, steps 100-104 that show the details of posting the send message to the NIC 84; column 7, lines 14-45 disclose the same details).

Consider **claim 3**, and **as it applies to claim 1 above**, Osborne further shows and discloses the claimed system wherein the buffer command information comprises a command to describe pinned-down memory buffers of the host (Fig. 1, pinned-down message buffers 62-64 of sender (host) 50 that store the Send command with its parameters (such as Source Address that points to the message’s content) and the message content copied from the Application 1 memory 66; column 7, lines 25-31 teach

the same details, further disclosing that the copying step is often optimized by mapping locations in the application memory 66 to locations in the message buffer 62 to avoid actual copying).

Consider **claim 4**, and as it applies to claim 3 above, Osborne further discloses the claimed system wherein the buffer command information comprises a command to bind a portion of the pinned down memory buffers of the host to a steering tag (STag) (column 7, lines 25-31 which further disclose that the copying step is often optimized by mapping locations (using pointers, considered by the examiner to correspond to applicants' steering tag STag, to de-reference the memory location) where the message content (i.e. payload) is actually stored in the application memory 66 to locations in the message buffer 62, so as to avoid actual copying of the content in the message buffers 62-64).

Consider **claim 5**, and as it applies to claim 1 above, Osborne further discloses the claimed system wherein the send command is an RMDA send message (Fig. 1, "Send" command 67 with parameters ID (identifying the receiver of the message), Source Address (location where the message content is stored), and Size (amount of data to be sent) stored in the buffer 66; column 7, lines 19-25 disclose the same details; and column 7, lines 37-45 which disclose using a direct memory access (DMA) with the network interface to send the message to message buffer 74 of the remote host 52).

Consider **claim 10**, and **as it applies to claim 1 above**, Osborne shows and discloses the claimed system, wherein the buffer command information provides a description of a section of memory (Fig. 1, Send command 67 with the parameter "Source Address" that points to a memory location in the application memory 66, where the content of the message being sent by RDMA is stored, thereby providing a description of a section of memory; the corresponding content when copied into the message buffers 62-64 of the operating system 56, also show that the buffer command information provides a description of a section of memory; column 7, lines 14-45 describe the same details).

Consider **claim 16**, and **as it applies to claim 1 above**, Osborne shows and discloses the claimed system wherein the NIC comprises an RDMA-enabled NIC (Fig. 1 that shows Network Interface (NIC) 84; column 7, lines 37-45 describe the NIC as a Remote DMA-enabled NIC).

Claims 6-9, 11 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Osborne (U.S. Patent Publication # 5,790,804)** in view of **Roach et al. (U.S. Patent Publication # 6,304,910 B1)**.

Consider **claim 6**, and **as it applies to claim 4 above**, Osborne discloses the claimed system, except wherein the NIC places the STag value in an optional field in a direct data placement DDP or RDMA header.

In the same field of endeavor, Roach et al. show and disclose the claimed system wherein the NIC places the STag value in an optional field in a direct data placement DDP or RDMA header (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values and BLM flag fields indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to place the STag value in an optional field in a direct data placement DDP or RDMA header, as taught by Roach et al., in the system of Osborne, so as to be able to handle non-contiguous data blocks during the transfer.

Consider **claim 7**, and as it applies to **claim 6 above**, Osborne discloses the claimed system, except wherein the NIC encodes a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid.

In the same field of endeavor, Roach et al. show and disclose a system wherein the NIC encodes a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values and BLM flag fields indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to encode a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid, as taught by Roach et al., in the system of Osborne, so as to be able to distinguish valid data block entries for remote transfer.

Consider **claim 8**, and as it applies to **claim 6 above**, Osborne discloses the claimed system, except wherein the NIC sets one or more bits in a field in the DDP or RDMA header indicating that the STag value in the optional field is valid.

In the same field of endeavor, Roach et al. show and disclose a system wherein the NIC sets one or more bits in a field in the DDP or RDMA header indicating that the STag value in the optional field is valid (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values and BLM flag fields indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to set a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid, as taught by Roach et al., in the system of Osborne, so as to be able to distinguish valid data block entries for remote transfer.

Consider **claim 9**, and **as it applies to claim 6 above**, Osborne discloses the claimed system, except wherein the NIC sets one or more bits or encodes a value into a second field in the DDP or RDMA header to advertise the portion of the pinned memory buffers of the host associated with the STag.

In the same field of endeavor, Roach et al. show and disclose a system wherein the NIC sets one or more bits or encodes a value into a second field in the DDP or RDMA header to advertise the portion of the pinned memory buffers of the host associated with the STag (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values and BLM flag fields indicating validity of the BSWA entries, thereby advertising the portion of the pinned memory buffers of the host associated with the STag).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to set one or more bits or encode a value into a second field in the DDP or RDMA header to advertise the portion of the pinned memory buffers of the host associated with the STag, as taught by Roach et al., in the system of Osborne, so as to be able to distinguish valid data block entries for remote transfer.

Consider **claim 11**, and **as it applies to claim 1 above**, Osborne discloses the claimed system, except wherein the single command message is posted to a command ring of the host.

In the same field of endeavor, Roach et al., disclose a system wherein the single command message is posted to a command ring of the host (Fig. 5A, block 32; column 6, lines 48-64 that disclose a command ring structure as a circular queue of command entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to post the single command message to a command ring of the host, as taught by Roach et al., in the system of Osborne, so as to be able to execute the commands in the FIFO order with opportunity for the non-serviced commands to get multiple turns at the service.

Consider **claim 24**, Osborne shows and discloses a method for transferring data over an RDMA network (Figs. 1 and 2 that show a method used with an application 58, at the sender 50 and receiver 52 wherein the sender and receiver communicate by sending messages to each other across the network 82; column 7, lines 14-45 disclose the details of the method, including reciting that the operating system may set up a direct memory access (DMA) with the network interface to extract and copy the message sent from sender's application 1 to message buffer 74 of the receiver's application 1), comprising:

initiating an RDMA write operation using a one-shot initiation process between a driver and a NIC of a host (Fig. 1 that shows a host (sender 50) with operating system 56, processor (driver) 54, and a network interface 84 that is coupled to the processor; column 7, lines 14-45 disclose the details of the method; Figs. 1-2 that further show the

method and list the steps necessary (in Fig. 2) to initiate, by the processor 54 and NIC 84 of host 50, the transfer of a message data from application 1 to the corresponding application 1 at the receiving host 52; column 7, lines 18-36 describe the steps shown in Fig. 2, starting at step 99, by invoking a "Send" command 67 (details shown in Fig. 1) by application 1, which is copied over from the application memory to processor message buffer 62 in step 100, and processing, by processor 54, the message content to a format compatible with the protocol being used in step 102, before handing over the initiation process to NIC 84, that sends the message and its content over the network 82 in step 104, for delivery to a remote receiving host 52; column 7, lines 37-45 further disclose using a direct memory access (DMA) with the network interface to extract and copy the message to message buffer 74 of the remote host 52); wherein the one-shot initiation process comprises communicating a single command message comprising buffer command information comprising commands to insert an STag value, and a write command to write an RDMA send message (Fig. 1 that shows a one-shot initiation process that comprises communicating a single Send command 67 message with parameters ID (of destination host), source address pointer (STag value that is part of the buffer command information), and message size (also a part of the buffer command information) from a sending host 50 to a receiving host 52 via a processor 54, NIC 84, network 82 to NIC 86, processor 70, and finally to receiving host 52), Fig. 1 further showing a copying process to copy the send command and the associated parameters of the send command from the Application 1 buffers 66 to the

operating system's message buffers 62-64, thereby disclosing a write command to write an RDMA send message).

However, Osborne does not show or disclose validating STag value; inserting an STag value in a first field of a DDP or RDMA header of the RDMA send message; and validating the STag value in the first field with a bit flag or other specified value in a second field of the DDP or RDMA header.

In the same field of endeavor, Roach et al. show and disclose inserting an STag value in a first field of a DDP or RDMA header of an RDMA send message (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values); and validating the STag value in the first field with a bit flag or other specified value in a second field of the DDP or RDMA header (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BLM flag fields indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to insert the STag value in a first field of a DDP or RDMA header of an RDMA send message, and validate the STag value in the first field with a bit flag or other specified value in a second field of the DDP or RDMA header, as taught by Roach et al., in the method of Osborne, so as to be able to handle non-

contiguous data blocks during the transfer and distinguish valid data block entries for remote transfer.

Claims 12-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Osborne (U.S. Patent Publication # 5,790,804)** in view of **Roach et al. (U.S. Patent Publication # 6,304,910 B1)** and further in view of **Tillier (U.S. Patent Publication # 6,421,742 B1)**.

Consider **claim 12**, and as it applies to **claim 11 above**, Osborne as modified by Roach et al., discloses the claimed system, except wherein the driver allocates an STag value.

In the same field of endeavor, Tillier shows and discloses the claimed system wherein the driver allocates an STag value (Fig. 8, arrow marked 'Pointer to "A"' which the examiner has interpreted to be equivalent to a STag value, being allocated by the Input/output unit (driver) and received by NIC (functionally represented by the middle column), the pointer value representing the address of the pinned down memory).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a system wherein the driver allocates an STag value, as taught by Tillier, in the system of Osborne, as modified by Roach et al., so as to associate and locate the pinned-down memory buffers using the STag value.

Consider **claim 13**, and **as it applies to claim 12 above**, Osborne as modified by Roach et al., and Tillier, further discloses the claimed system, wherein the STag value is returned synchronously from a command call (in Roach et al. reference, column 6, lines 56-64 which disclose that the host driver increments the “put” pointer whenever a command is queued to the command ring, making the updated pointer (STag value) synchronously available from a call command).

Consider **claim 14**, and **as it applies to claim 12 above**, Osborne, as modified by Roach et al., and Tillier, further discloses the claimed system, wherein the STag value is saved in a driver command table of the host (in Roach et al. reference, Fig. 3, host memory block 42 and transfer ready queue 60; column 5, lines 44-52 that disclose a lookup field (STag value) inside each frame header includes a pointer to an associated context, and the host driver saves these fields in the host memory 42).

Consider **claim 15**, and **as it applies to claim 14 above**, Osborne, as modified by Roach et al., and Tillier, further discloses the claimed system, wherein the STag value saved in a driver command table is associated with an application reference number (in Roach et al. reference, Figs. 5A and 5B; column 5, lines 47-52 which disclose that the context field associated with a pointer field (STag value) saved in a driver command table is associated with a small computer systems interface (SCSI) state information interpreted by the examiner to be associated with the application reference number).

Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya (U.S. Patent Publication # 7,376,755 B2)**.

Consider **claim 17**, Osborne shows and discloses a system for transferring data over a remote direct memory access (RDMA) network (Figs. 1 and 2 that show a system with an application 78, at the receiver host 52, that communicates with the corresponding application 1 at the sender host 50, by receiving/sending messages to each other across the network 82; column 7, lines 14-45 disclose the details of the system, including reciting that the operating system may set up a direct memory access (DMA) with the network interface to extract and copy the message sent from sender's application 1 to message buffer 74 of the receiver's application 1), comprising: a host comprising a driver and a network interface card (NIC), the driver being coupled to the NIC (Fig. 1 that shows a receiving host 52 with operating system 72, processor (driver) 70, and a network interface 86 that is coupled to the processor; column 7, lines 14-45 disclose the details of the system), wherein a one-shot completion process of an RDMA operation is performed between the driver and the NIC of the host (Figs. 1-2 that show the system and list the steps necessary (in Fig. 2) to complete, by the processor 70 and NIC 86 of the receiving host 52, the transfer of a message data from application 1 at the sender host 50 to the corresponding application 1 at the receiving host 52, and a "Receive" command 77

(details shown in Fig. 1); column 7, lines 37-65 describe the message receiving steps 106-110 shown in Fig. 2, further disclosing that the message arrival at the receiver causes an interrupt to the processor 70, and the operating system 72 extracts the message (step 106) from the network in cooperation with NIC 86, and stores it into message buffer 74 in the operating system using Direct Memory Access (DMA); the operating system 72 then performs protocol processing, if necessary, in step 108; then copies the message in step 110 from the message buffer 72 at the receiver 52 to application memory (endpoint 79), in response to a "Receive" command request 77 from application 78).

However, Osborne does not explicitly disclose that the one-shot completion process comprises communicating a single completion message comprising: a send complete indication, and buffer freeing status information.

In the same field of endeavor, Pandya discloses the claimed system, including sending a send complete indication, and buffer freeing status information (Fig. 33 that shows the details of a "Write" operation, including send status and sense step 3310 at the completion of the "Send Data" operation; column 13, lines 35-48 which describe that at the completion of data transfer (associated with the send data command), the status of the command execution is passed onto the host SCSI layer, which then does the appropriate processing, including releasing the buffers being used for data transfer to the application).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to include a send complete indication, and buffer

freeing status information, as taught by Pandya, in the system of Osborne, so as to enable the sending host to perform clean-up operations after the message with its content has been successfully received by the receiving host.

Claims 18, 20, 22 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Osborne (U.S. Patent Publication # 5,790,804)** in view of **Pandya (U.S. Patent Publication # 7,376,755 B2)** and further in view of **Tillier (U.S. Patent Publication # 6,421,742 B1)**.

Consider **claim 18, and as it applies to claim 17 above**, Osborne, as modified by Pandya, shows and discloses the claimed system, except wherein the NIC receives a message comprising an optional field carrying a STag value, the STag value being associated with pinned memory in a remote host.

In the same field of endeavor, Tillier shows and discloses the claimed system, wherein the NIC receives a message comprising an optional field carrying an STag value, the STag value being associated with pinned memory in a remote host (Fig. 8, arrow marked 'Pointer to "A"' which the examiner has interpreted to be equivalent to a STag value, being received by NIC along with the "Store Data" command, the pointer value representing the address of the pinned down memory in a remote host).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide a system wherein the NIC receives a message comprising an optional field carrying an STag value, the STag value being

associated with pinned memory in a remote host, as taught by Tillier, in the system of Osborne, as modified by Pandya, so as to be able to access the message content in the pinned memory in the remote host.

Consider **claim 20**, and as it applies to claim 18 above, Osborne, as modified by Pandya and Tillier, further shows and discloses a system wherein the NIC de-associates the STag value with the pinned memory in the host, thereby preventing further access to the pinned memory using the de-associated STag value (in Tillier reference, Fig. 6, blocks 604-606 that depict freeing up of allocated resources, corresponding to de-associating the STag value from the pinned-memory in the host, thereby preventing further access to the pinned memory using the de-associated STag value; column 8, lines 9-15 that disclose the same details).

Consider **claim 22**, and as it applies to claim 18 above, Osborne, as modified by Pandya and Tillier, further shows and discloses the claimed system wherein the NIC de-associates the STag value with previously associated SGL information (in Tillier reference, Fig. 6, blocks 604-606 that depict freeing up of allocated resources, corresponding to de-associating the STag value pointing to the SGL; cleanup routine of Fig. 7B; column 8, lines 9-15 that disclose the same details).

Consider **claim 23**, and as it applies to claim 20 above, Osborne, as modified by Pandya and Tillier, further shows and discloses a system wherein the NIC frees any

resources dedicated to information regarding the pinned memory (in Tillier reference, Fig. 6, blocks 604-606 that depict freeing up of allocated resources; cleanup routine of Fig. 7B; column 8, lines 9-15 that disclose NIC freeing up the allocated resources).

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Osborne (U.S. Patent Publication # 5,790,804) in view of Pandya (U.S. Patent Publication # 7,376,755 B2) and further in view of Tillier (U.S. Patent Publication # 6,421,742 B1) and further in view of Roach et al. (U.S. Patent Publication # 6,304,910 B1).

Consider **claim 19**, and as it applies to **claim 18 above**, Osborne, as modified by Pandya and Tillier, discloses the claimed system, except wherein a header of the message indicates the validity of the optional field with a bit flag or specified value in an encoded field.

In the same field of endeavor, Roach et al. show and disclose a system wherein a header of the message indicates the validity of the optional field with a bit flag or specified value in an encoded field (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values and BLM flag fields indicating validity of the BSWA entries, thereby indicating the validity of the optional field with a bit flag or specified value in an encoded field).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to indicate the validity of the optional field with a bit flag or specified value in an encoded field, as taught by Roach et al., in the system of Osborne, as modified by Pandya and Tillier, so as to be able to distinguish valid data block entries for remote transfer.

Claim 21 is rejected under 35 U.S.C. 103(a) as being unpatentable over Osborne (U.S. Patent Publication # 5,790,804) in view of Pandya (U.S. Patent Publication # 7,376,755 B2) and further in view of Tillier (U.S. Patent Publication # 6,421,742 B1) and further in view of Futral et al. (U.S. Patent Publication # 5,991,797).

Consider **claim 21**, and as it applies to **claim 18 above**, Osborne, as modified by Pandya and Tillier, further discloses the claimed system, wherein the single completion message comprises the optional field carrying the STag value received by the NIC; and wherein the NIC delivers the single completion message to the driver (in Pandya reference, Fig. 35, Target Operations 3507-3509 that send "Status and Sense" message via NIC and Driver to the Initiator of "Write" command at the completion of the command; since the STag field is optional, it is not shown; the NIC then delivers the message to the LAN driver 716 shown in Fig. 7; column 34, lines 29-30 disclose the response to data transfer completion operation).

However, Osborne, as modified by Pandya and Tillier, do not specifically show or disclose the claimed system wherein the driver compares the STag value received with a STag value previously sent.

In the same field of endeavor, Futral et al. show and disclose a system wherein the NIC delivers the message to the driver, and wherein the driver compares the STag value received with a STag value previously sent (Fig. 4, SGL word 2 (chain pointer) that provides address to additional transaction detail element list, when the buffers are scattered across multiple locations, therefore requiring an appended list. In such cases, the driver compares the SGL address received with a prior value to ensure that the same list is being processed with additional buffer addresses in the appended chain list; column 6, lines 36-46 discloses the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide capability in the driver to compare the STag value received with a STag value previously sent, as taught by Futral et al., in the system of Osborne, as modified by Pandya and Tillier, so as to be able to handle multiple non-contiguous data blocks during the transfer.

Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Osborne (U.S. Patent Publication # 5,790,804) in view of Pandya (U.S. Patent Publication # 7,376,755 B2) and further in view of Roach et al. (U.S. Patent Publication # 6,304,910 B1).

Consider **claim 25**, Osborne shows and discloses a method for transferring data over an RDMA network (Figs. 1 and 2 that show a method used with an application 58, at the sender 50 and receiver 52 wherein the sender and receiver communicate by sending messages to each other across the network 82; column 7, lines 14-45 disclose the details of the method, including reciting that the operating system may set up a direct memory access (DMA) with the network interface to extract and copy the message sent from sender's application 1 to message buffer 74 of the receiver's application 1), comprising:

completing an RDMA write operation using a one-shot completion process between a NIC and a driver of a host (Figs. 1-2 that show the system and list the steps necessary (in Fig. 2) to complete, by the processor 70 and NIC 86 of the receiving host 52, the transfer of a message data from application 1 at the sender host 50 to the corresponding application 1 at the receiving host 52, and a "Receive" command 77 (details shown in Fig. 1); column 7, lines 37-65 describe the message receiving steps 106-110 shown in Fig. 2, further disclosing that the message arrival at the receiver causes an interrupt to the processor 70, and the operating system 72 extracts the message (step 106) from the network in cooperation with NIC 86, and stores it into message buffer 74 in the operating system using Direct Memory Access (DMA); the operating system 72 then performs protocol processing, if necessary, in step 108; then copies the message in step 110 from the message buffer 72 at the receiver 52 to application memory (endpoint 79), in response to a "Receive" command request 77 from application 78).

However, Osborne does not specifically show and disclose that the one-shot completion process comprises communicating a single completion message comprising:

a send complete indication, buffer freeing status information, and an STag value; receiving the single completion message; identifying the STag value in a first field of a header of the single completion message; and validating the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header.

In the same field of endeavor, Pandya shows and discloses the claimed method, wherein the one-shot completion process comprises communicating a single completion message comprising:

a send complete indication, buffer freeing status information, and an STag value (Fig. 33 that shows the details of a "Write" operation, including send status and sense step 3310 at the completion of the "Send Data" operation; column 13, lines 35-48 which describe that at the completion of data transfer (associated with the send data command), the status of the command execution is passed onto the host SCSI layer, which then does the appropriate processing, including releasing the buffers being used for data transfer to the application; since releasing the buffers implies knowing where in the memory the buffers are, it is implied that a pointer (STag) to the message buffers is also sent with the send complete indication message); and receiving the single completion message (Fig. 33, message "send status and sense" 3310 from the target to the initiator in response to the "Write" command request 3301,

wherein the message indicates command complete message 3312 at the initiator; column 13, lines 35-48 which describe the same details).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to provide in the one-shot completion process that comprises communicating a single completion message, a send complete indication, buffer freeing status information, and an STag value, as taught by Pandya, in the method of Osborne, so as to free up the locked (pinned-down) buffers after the data transfer between the two hosts is complete.

However, Osborne, as modified by Pandya, does not disclose identifying a STag value in a first field of a header of the completion message; and validating the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header.

In the same field of endeavor, Roach et al. show and disclose identifying a STag value in a first field of a header of the completion message (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BSWA buffer pointer list entries corresponding to the STag values); and validating the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header (Fig. 8 that shows the bit-level definition of each of the two 32-bit word entry shown in Fig. 9; column 8, lines 32-67 and column 9, lines 1-18 that disclose the details of the bit-level structure including BLM flag fields indicating validity of the BSWA entries).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to identify a STag value in a first field of a header of the completion message, and validate the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header, as taught by Roach et al., in the method of Osborne, as modified by Pandya, so as to be able to handle non-contiguous data blocks during the transfer and distinguish valid data block entries for remote transfer.

(10) Response to Argument

The appellants have presented eight different grounds for review by the Board of Patent Appeals and Interference. The examiner would like to summarize the issues raised in each ground under review and then present his response to each of the issues raised.

Ground I: The 35 U.S.C. 102 (b) rejection of claims 1-5, 10, and 16

Ground I includes **subsection A** (arguments against rejection of independent **claim 1**) and **subsection B** (arguments against rejection of **dependent claims 2-5, 10 and 16**).

Ground I-A: On page 10 of appeal brief, the appellants argue that the cited reference of Osborne does not disclose the details of the initiation process of a one-shot RDMA comprising communicating a single command message that includes buffer command information, and a write command to write a send command.

On page 11 of the appeal brief, the appellants argue that nowhere in Osborne is there any specific disclosure regarding anything being sent between Osborne's processor 54 and network interface 84.

In the last paragraph on page 11 of the appeal brief, the appellants allege that Osborne's "Send" command does not contain any buffer command information.

On page 12 of the appeal brief, the appellants state that "it is the message data copied/mapped from the application memory 66 to operating system message buffer 62 that is sent over network 82 via network interface 84, not the send command 67 sent between application 58 and operating system 56".

Ground I-B: On page 14 of appeal brief, the appellants argue that “copying a send command” as recited in claim 2 is different from “copying message data” disclosed by Osborne. The appellants then allege that Osborne clearly cannot disclose “wherein the driver posts the single command message to perform the one-shot initiation process”, because nowhere in Osborne is there any disclosure regarding the functions of the processor 54.

In the last paragraph on page 14 of the appeal brief, the appellants further argue that “mapping locations in the application memory to message buffers does not pin-down memory buffers of the host”, as recited in claim 3.

On page 15 of the appeal brief, the appellants, with reference to claim 4, assert that nowhere in Osborne are steering tags disclosed. Therefore, Osborne clearly fails to disclose “wherein the buffer command information comprises a command to bind a portion of the pinned-down memory buffers of the host to a steering tag (STag)”, as recited in appellants’ claim 4.

On page 16 of the appeal brief, referring to claim 5, the appellants repeat the previous argument that the send message of Osborne, including a receiver ID, source address and size of message data does not teach an RDMA send message, as disclosed in claim 5.

On page 16 of the appeal brief, the appellants argue that performing DMA operations do not correspond to RDMA operations across a network, as recited in appellants’ dependent claim 16.

Examiner's Response:

Ground I-A: The examiner respectfully disagrees with the appellants' argument that the cited reference of Osborne does not disclose the details of the initiation process of a one-shot RDMA comprising communicating a single command message that includes buffer command information, and a write command to write a send command. It is said that a picture is worth a thousand words. In the cited Osborne prior art, Fig. 1 shows all the claimed details of the initiation process of a one-shot RDMA, by setting up a single (i.e. one-shot) "send" command that includes a first parameter "ID" providing the destination address of the receiving node 52, a second parameter "Source Address" which is a pointer to a buffer or buffers that hold the message data to be transferred to the receiving node 52, and a third parameter "Size" that gives the amount of data to be transferred. The setting up of the send command and initializing the values of the three parameters of the command before the actual transfer of the message data corresponds to the initiation process recited in claim 1.

As Fig. 1 clearly shows, the sending node 50 and the receiving node 52 are remote to each other, connected by network 82. Further, as disclosed in the cited Osborne reference (in column 7, lines 15-45), the operating system may set up a Direct Memory Access (DMA) with the network interface 84 to transfer the message data across network 82 to the memory buffer 74 of the receiving node 52. Thus, Osborne reference does disclose the details of the initiation process of a one-shot RDMA.

Furthermore, Osborne reference (in column 7, lines 25-27) teaches copying (i.e. writing) the send message from application memory 66 to the operating system memory

62 of the sending node. The examiner has therefore shown that Osborne teaches every claim element related to the initiation process of a one-shot RDMA recited in appellants' claim 1.

The examiner begs to differ with the appellants' argument that nowhere in Osborne is there any specific disclosure regarding anything being sent between Osborne's processor 54 and network interface 84. By looking at the direction of arrows (representing flow of data) in Fig. 1, one would notice that data flows from application 1 at the sending node via processor 54 (driver) of the operating system 56 and NIC 84, across network 82 to the NIC 86 of the receiving node 52, and then via processor 70 (driver) of the operating system 72 to the memory of application 1 of the receiving node 52. That is what an RDMA operation involves.

As to the appellants' allegation that Osborne's "Send" command does not contain any buffer command information, the examiner would like to refer to the second parameter "Source Address" (which is a pointer to the message buffer) and the third parameter "Size" (that indicates the amount of data in the message buffer that is to be transported by the RDMA operation) described above. The appellants' argument is therefore without any merit.

The examiner cannot disagree more with appellants' assertion that it is the message data copied/mapped from the application memory 66 to operating system message buffer 62 that is sent over network 82 via network interface 84, not the send command 67 sent between application 58 and operating system 56. As a person of ordinary skill in the art would know that in a network, data does not get transmitted

without any command, destination address, and other applicable parameters. Every data packet has a header that includes a command followed by associated parameters. The NIC 84 translates the "Send" command shown in Fig. 1 to TCP-IP specific binary command that the other nodes in the network 82 understand and forward the message data across the network to the destination NIC 86.

In conclusion, the cited Osborne reference does teach all the claimed elements recited in the independent claim 1, which is, therefore, fully anticipated by Osborne, and is not in a condition for allowance.

Ground I-B: In response to the appellants argument that "copying a send command", as recited in claim 2, is different from "copying message data" disclosed by Osborne, the examiner states that the message data includes both the command and the data (payload). As explained earlier, every data packet must include a header with at least one command and any associated parameters, otherwise, payload by itself cannot be transported across the network.

The appellants' allegation that Osborne clearly cannot disclose "wherein the driver posts the single command message to perform the one-shot initiation process", because nowhere in Osborne is there any disclosure regarding the functions of the processor 54, is without any merit, because the processor 54 interfaces with NIC 84. As one of the interface functions, the processor 54 issues commands to NIC 84 to communicate with remote nodes. The "Send" command shown in Fig. 1 of Osborne, and disclosed in column 7, lines 15-45, is one such command that initiates transfer of message data to the remote node 52. The operating system 56, in the Osborne

reference, provides machine-coded instructions that are executed by the hardware of the processor 54 to post a single “Send” command to NIC 84, as recited in claim 2 of the appellants.

The examiner further disagrees with the appellants’ argument that “mapping locations in the application memory to message buffers does not pin-down memory buffers of the host”, as recited in claim 3. When message buffers are allocated and mapped in the host memory as shown in Fig. 1, those memory locations are reserved for the duration of the operation (i.e. pinned down) until specifically released by the operating system. Thus Osborne shows and provides sufficient disclosure to teach claim 3 element, which recites that “the buffer command information comprises a command to describe pinned-down memory buffers of the host”.

The appellants’ assertion that Osborne does not disclose any steering tag is also without any merit. The steering tag/STag is used as a pointer that holds address of a buffer (please refer to appellants’ provisional application 60/404,709 to page 3, paragraph 09, item b) which defines STag as “NIC builds a mapping table for the pinned down buffers on Host 1. This mapping table is given a handle (i.e. a pointer) called an STag”, which compares well with Osborne’s recitation in column 7, lines 25-31 that teach “The operating system then copies the message from application memory to message buffers 62, in the operating system 56 of sender 50. This step is often optimized by mapping locations in the application memory to locations in the message buffer to avoid actual copying.”). Osborne’s “Send” message clearly includes such a pointer in the form of “Source Address” parameter that references (binds) a portion of

the pinned-down (reserved) memory buffers of the host, as recited in appellants' claim 4.

In response to the appellants' argument that the send message of Osborne, including a receiver ID, source address and size of message data does not teach an RDMA send message, as disclosed in claim 5, the examiner would like to repeat that a person of ordinary skill in the art would readily be able to understand, by looking at Fig. 1 of Osborne and reviewing the disclosure in column 7, lines 15-45 that the send command is indeed an RDMA send message.

The examiner disagrees with the appellants' argument that performing DMA operations across network do not correspond to RDMA operations across a network, as recited in appellants' dependent claim 16. Fig. 1 clearly shows and column 7, lines 15-45 disclose that nodes 50 and 52 are remote from each other, connected via network 82, and perform a DMA (Direct Memory Access) operation between the two nodes 50 and 52. Therefore, it does not matter that nowhere in Osborne is there any mention of "remote direct memory access" or "RDMA", so long as one of the ordinary skill in the art can conclude that Osborne does teach such an operation.

In conclusion, claims 2-5, 10 and 16 are fully anticipated by Osborne, and are therefore not allowable.

Ground II: Claims 6-9, 11, and 24 are not obvious over Osborne in view of Roach

Ground II includes **subsection A** (arguments against rejection of independent claim 24) and **subsection B** (arguments against rejection of dependent claims 6-9 and 16).

Ground II-A: On pages 17-21 of appeal brief, the appellants repeat the same arguments for the method claim 24 that were presented for the system claim 1 in section I-A. The examiner has already responded to each repeated argument in the "Examiner's Response" section, Ground 1-A for claim 1. The same response is applicable for claim 24.

On page 22 of the appeal brief, the appellants argue that nowhere in Roach (cited secondary reference) is there any disclosure of STag values, and nothing in Roach teaches that the fiber channel frames are RDMA send messages.

Ground II-B: On page 25 of the appeal brief, the appellants merely claim that the cited reference of Roach et al fails to teach claim 6 element that recites "the system wherein the NIC places the STag value in an optional field in a direct data placement (DDP) or RDMA header"; and that Roach et al. also fail to disclose "the system wherein the NIC encodes a value into a field in the DDP or RDMA header indicating that the STag value in the optional field is valid", as recited in claim 7; and that Roach et al. further fail to disclose "the system wherein the NIC sets one or more bits in a field in the DDP or RDMA header indicating that the STag value in the optional field is valid", as recited in claim 8; and further claiming that Roach et al. do not disclose "the system wherein the NIC sets one or more bits or encodes a value into a second field in the DDP

or RDMA header to advertise the portion of the pinned down memory buffers of the host associated with the STag", as recited in claim 9.

Examiner's Response:

Ground II-A: In response to the appellants' argument that the cited reference of Roach et al. does not describe any STag values, the examiner would like to state that STag is a term used by the appellants for a pointer to locate data stored at specified (value of the pointer) memory location (please see Examiner's Response for Ground 1-B). As can be clearly seen in Figs. 8-9 of Roach et al., a series of pointers (BSWA or Buffer Starting Word Address representing STag value) are used as buffer list entry pointers. In Fig. 8, Roach further shows the bit-level details of each such pointer entry (two 32-bit words), wherein the first word (word 0) is the pointer that holds the address of a buffer storing message data, and the second word (word1) stores control information including BLM (Buffer List Modifiers) field, the bit-level details of which are listed in column 8, lines 44-67 through column 9, lines 1-18. Some of these bits (e.g. bits 31, 28-25) are used for buffer pointer validation purposes.

The examiner disagrees with the appellants' argument that Roach et al. do not teach RDMA operation. Roach does teach DMA operation (for example, column 4, lines 53-63 and several other paragraphs). Since these operations involve data transfer between network nodes separated across a network, RDMA operation is taught in the Roach et al. reference, even if not explicitly mentioned as RDMA. In any case, Osborne reference already teaches RDMA. Roach et al. reference was combined with the

teachings of Osborne to describe features corresponding to “inserting STag value in a first field of a DDP or RDMA header of the RDMA send message,” and “validating the STag value in the first field with a bit flag or other specified value in a second field (e.g. BLM) of the DDP or RDMA header”. Even if Roach used FC-2 header (Fiber Channel frame header) to describe the insertion of STag pointers and setting BLM bits to validate the STag pointers, the concept may be applied to an RDMA header as well.

Accordingly, claim 24 is obvious over Osborne in view of Roach et al., and therefore not allowable.

Ground II-B: The examiner contends that Roach et al. show and disclose all the claim elements of dependent system claims 6-9 in Figs. 8-9 and column 8, lines 32-67 through column 9, lines 1-18. For example, Fig. 9 shows a fiber channel header with reserved fields and optional fields as well as undefined fields, wherein the optional fields are used to place BSWA pointer (STag) values in them (recited in claim 6), and to encode (recited in claim 7) and set (recited in claim 8) the bit values in the BLM field to indicate validity of the BSWA pointers; and by setting one or more bits or encoding a value for the set bits, advertises (recited in claim 9) the corresponding buffers addressed by the BSWA pointers to be valid (e.g. Bit 31 when set to 1, indicates receive only buffer, but when set to a '0'b, may be used for transmit buffers). Fig. 3 shows a host 40 interfacing with a communication processor 22 that incorporates receive and transmit protocol engines 30 and 32 respectively and NL-Port 36 (all part of NIC functions); column 4, lines 53-63 disclose the same details.

Therefore, dependent claims 6-9 and 11 are obvious over the teachings of Osborne, in view of Roach et al., and not in the condition for allowance.

Ground III: Claims 12-15 are not obvious over Osborne in view of Roach and further in view of Tillier

Ground III: On page 26 of the appeal brief, the appellants argue that the cited secondary reference of Tillier does not teach the claimed feature "wherein the driver allocates an STag value", as recited in dependent claim 12. The appellants also continue to argue that nowhere in Tillier reference is there any mention of the term "steering tag/STag".

On page 27 of the appeal brief, the appellants repeat the argument that the cited Roach et al. reference does not disclose STag values, emphasizing that the well known terms "steering tag" and "STag" do not even appear in Roach. The appellants also assert that "as one of ordinary skill in the art would readily understand, STag values are not incremented", and since Roach's put and get pointers are incremented as commands are added to the command ring, they cannot be STag values, wherein the dependent claim 13 specifies STag value that is not incremented.

Examiner's Response:

Ground III: The examiner has already responded to the appellants' argument about the use of the term "steering tag/STag" in Ground 1-B. As to the appellants' argument that Tillier does not allocate the pointer because Tillier's I/O unit receives the

pointer from the host, the examiner would like to state that the Tillier reference was added to merely show and disclose a device **driver 103-2** (see Fig. 1), because the cited Osborne reference does not specifically refer to the operating system 56 modules and processor 54 as driver recited in claim 12. Otherwise, Osborne clearly shows allocation of a pointer (column 7, lines 25-31) that corresponds to the "source address" parameter of the "Send" command shown in Fig. 1. Furthermore, the argument that Tillier reference only receives but does not allocate an STag is also incorrect. As shown in Fig. 8 of Tillier, there are two different pointers, one received from the Host (to the left in Fig. 8) along with the command, and the other (Pointer to "A") allocated by the I/O unit to store Data "A" at a location in storage device specified by the Pointer to "A".

The examiner begs to differ with the appellants' assertion (for claim 13) that STag values cannot be incremented. As used by the appellants, STag represents a pointer, whose value refers to an address of a buffer or memory location. Such pointer values are often incremented or decremented to point to new locations in memory. The reason Roach increases the put pointer is because a new command stored in the command ring buffer must be stored at a different address than the previous command's location in the buffer. Since the appellants' invention uses only a single command, there is no need to increase the pointer value. But, that does not mean that STag values cannot be incremented in any situation, as the appellants assert.

Accordingly, dependent claims 12-15 are obvious over Osborne in view of Roach et al. and Tillier, and therefore not allowable.

Ground IV: Claim 17 is not obvious over Osborne in view of Pandya

Ground IV: On page 29 of the appeal brief, the appellants point out that in the cited Pandya reference, an RDMA Write operation is shown in Fig. 35 instead of the examiner's cited Fig. of 33. The appellants further argue that in the cited column 13, lines 35-48, Pandya explicitly teaches that the "buffers are not freed until the status of the command execution is passed onto the host SCSI layer, which releases the buffers", then asserting that because the buffers are not freed until after the completion message has already passed between the NIC and the driver, the message between the driver and NIC cannot comprise buffer freeing status information. The appellants then argue that "sense status is merely error information, which is different than buffer freeing status information".

On page 30 of the appeal brief, the appellants mischaracterize what the examiner explained in the requested telephonic interview. The examiner's response to the appellant's argument in this appeal brief is shown below, and that should be the only response under review during this appeal, not what the appellants characterize as the examiner's response in the telephonic interview.

On pages 31-32 of the appeal brief, the appellants state that "one of ordinary skill in the art would clearly not be motivated to combine Osborne's teaching with that of Pandya's, because Osborne's teachings occur during data transfer at a receiver, versus Pandya's teachings after data transfer at a receiver.

Examiner's Response:

Ground IV: The examiner would like to refer to Fig. 33 in the Pandya reference, because the Pandya invention is titled "TCP/IP Processor and Engine using RDMA". Therefore, the entire invention uses RDMA and either of the two figures (33 or 35) will suffice. The examiner chose Fig. 33, because it is simpler in operation. As to appellants' argument that in Pandya the buffers are not freed until the status of the command execution is passed onto the host SCSI layer, which releases the buffers, which is different than the claimed feature of "completion message comprising a send completion indication, and buffer freeing status information", the examiner would like to clarify that once the receiving node has copied all the transmitted message data successfully, it frees its own buffers and sends a normal completion status (such as a return code set to zero) and suggesting to the sending node to free its own buffers. The receiving node cannot free the allocated buffers (resources) of the sending node, since it has no such access rights, it can only indicate the status of its own buffers that have been freed, after the normal transfer of message data. The examiner also disagrees with the appellants' assertion that "as is well known in the art, sense status is merely error information" that cannot represent "send completion indication" and "buffer freeing status information" features of claim 17. What is well known in the art is that a return code with a zero value generally represents normal completion of an operation, while a non-zero value represents a code in a look-up table for a plurality of abnormal ending (abend) operations.

In response to the appellants' statement that Osborne's teachings cannot be combined with Pandya's teachings, the examiner would like to state that in a distributed

data processing environment involving processes that execute at different network nodes, the combined teachings of Osborne and Pandya are often used as an inter-process communication stage at the end of an operation.

Accordingly, independent claim 17 is obvious over Osborne in view of Pandya, and therefore not allowable.

Ground V: Claims 18, 20 and 22-23 are not obvious over Osborne in view of Pandya and further in view of Tillier

Ground V: On page 33 of the appeal brief, the appellants repeat the same arguments for dependent claim 18 that the examiner has already responded to in Ground III, namely Tillier does not disclose an STag, and the I/O unit is separate from Host of Fig. 8 in Tillier, and therefore not a NIC of a host; and the data stored at the host is not pinned down.

On page 34 of the appeal brief, the appellants repeat the same STag arguments for dependent claim 20 that they raised for dependent claim 18, then further argue that Tillier fails to teach that a NIC performs de-association of STag by freeing up the pinned down buffers.

On page 35 of the appeal brief, the appellants repeat the same STag arguments for dependent claim 22 that they raised for dependent claim 18, then further argue that Tillier fails to teach that the freeing of resources includes de-associating STag values from SGL information. The appellants then repeat the argument presented for claim 20 that Tillier fails to disclose that a NIC performs the de-association.

Also on page 35 of the appeal brief, the appellants continue to repeat the same arguments for dependent claim 23, for which the examiner has already provided response.

Examiner's Response:

Ground V: To summarize, as explained above, Osborne reference teaches a "source address" pointer parameter corresponding to STag value of claim 18, and since an allocated pointer represents address of a memory buffer reserved (i.e. pinned down) for an intended operation, and furthermore, since "source address" is not a commonly used and defined parameter for a generic packet header, it can only be placed in the optional field area of the packet header. Therefore, Osborne does teach the elements of claim 18 at the sending node, whereas Tillier shows and discloses the corresponding pointer (STag) at the receiving node.

In response to appellants' argument that Tillier does not show or disclose freeing up reserved resources (buffers) after the transfer of message data from the sending host to the receiving host is complete, as disclosed in dependent claim 20, the examiner would like to state that Tillier's Fig. 6, steps 604-606 and column 8, lines 9-15 clearly show and disclose the claimed features by stating that "before terminating, the buffers mapped in the system are unmapped (i.e. de-associated from the pointer) in step 604, the I/O request is freed in step 605, and the resources associated with the I/O request are freed. Fig. 7 shows corresponding pseudo code for the disclosed features.

As to the argument that Tillier fails to teach that a NIC performs de-association of STag, as recited in claim 20, the examiner would like to state that this feature is already disclosed in the cited Pandya reference (see Fig. 7 for the architecture of RDMA operations that shows NICs both at the sending and the receiving nodes, which when associated with the operations depicted in Fig. 33 and disclosed in column 13, lines 35-48, show that at the completion stage, the NICs free up the allocated buffers 705 and 710 at their respective nodes and the hosts free up their corresponding buffers 701 and 706).

The appellants' argument that Osborne, in view of Pandya and Tillier, fails to disclose that the freeing of resources includes de-associating STag values from SGL information, as recited in claim 22, is without any merit, because Osborne shows (in Fig. 1) a plurality of message buffers that can be referenced by a list of pointers (corresponding to Scatter Gather List or SGL when the message buffers are not in contiguous memory space); and column 7, lines 28-31 that disclose the mapping of the scattered message buffer addresses; and the Pandya as well as Tillier references that describe freeing up the reserved buffers (i.e. de-associating STag values from SGL information) after the message data transfer has been transferred successfully, as listed above.

In summary, the response to arguments for dependent claim 23 is that the pinned down memory is no more than a temporarily reserved memory buffer that is for the exclusive use of the operation for which it is reserved, until freed.

Ground VI: Claim 19 is not obvious over Osborne in view of Pandya, in further view of Tillier, and still further view of Roach

Ground VI: On page 36 of the appeal brief, the appellants argue that the dependent claim 19 is allowable because of its dependency on the independent claim 17, which they argue is allowable.

Examiner's Response:

Ground VI: Since the examiner has already responded to claim 17 arguments, and since the dependent claim 19 inherits all the limitations of its base claim 17, the examiner responds that, in the absence of any new argument, claim 19 is also considered not allowable.

Ground VII: Claim 21 is not obvious over Osborne in view of Pandya, in further view of Tillier, and still further view of Futral

Ground VII: On page 37 of the appeal brief, the appellants argue that the chain pointers within SGLs of the cited reference of Futral are not STag values.

On page 37 of the appeal brief, the appellants further argue that "nothing in Futral teaches comparing the chain pointers received to chain pointers previously sent, let alone using a driver to make the comparison".

Examiner's Response:

Ground VII: For the response to appellants' argument on page 37 of the appeal brief, please refer to the examiner's response in Ground 1-B for claim 4.

In response to the second argument on page 37 that Futral fails to teach any comparison by the driver, the examiner would like to refer to Fig. 4 in Futral, wherein two blocks of data are shown. The first block has a chain pointer 64 that points to (i.e. contains address of) the beginning of the second block. If there were additional blocks containing transaction detail elements (e.g. 4 total blocks instead of 2), the driver must compare the chain pointer address value with the other three blocks' starting address to maintain the order within the SGL list.

Accordingly, dependent claim 21 is obvious over Osborne in view of Pandya, Tillier and Futral, and therefore not allowable.

Ground VIII: Claim 25 is not obvious over Osborne in view of Pandya and further in view of Roach

Ground VIII: On page 38 of the appeal brief, the appellants repeat the argument of claim 17 for claim 25, that the combination of Osborne, Pandya and Roach does not disclose or suggest the limitation of "completing an RDMA write operation using a one-shot completion process between a NIC and a driver of a host, wherein the one-shot completion process comprises communicating a single completion message comprising: a send complete indication and buffer freeing status information" as set forth in appellants' independent claim 25.

On page 39 of the appeal brief, the appellants claim that their one-shot completion process communicates a single completion message.

For the arguments on pages 39-40 of the appeal brief, disputing whether Fig. 33 or 35 is the correct figure in Pandya reference, and Pandya's teachings in column 13, lines 35-48, please refer to the examiner's response in Ground IV rejection of claim 17 above.

On page 41 of the appeal brief, the appellants allege that the final Office action fails to provide rationale or evidence showing that an STag will necessarily need to be present in a completion message in order to release buffers.

For the argument presented on page 42 of the appeal brief, please refer to the examiner's response for claim 17 in Ground IV.

For the argument presented on page 43 of the appeal brief that the teachings of Osborne and Pandya references cannot be combined, please refer to the examiner's response for claim 17 in Ground IV.

On page 44 of the appeal brief, the appellants argue that nowhere in Roach is there any disclosure of STag values. The appellants also allege that "Roach is wholly unrelated to identifying an STag value in a first field of a header and validating the STag value in the first field of the header by identifying a bit flag or other specified value in a second field of the header".

Examiner's Response:

Ground VIII: Please refer to the examiner's response for Ground IV rejection of claim 17 above, for the argument presented on page 38.

Regarding the appellants' argument on page 39 of the appeal brief, Figs. 7 and 33 of Pandya reference clearly show that a single completion message is also conveyed between a NIC and a driver of a host in steps 3310-3312 of Fig. 33.

In response to appellants' allegation on page 41 of the appeal brief, the examiner has used Roach's teachings and not a conclusory statement to reject the claim element. Furthermore, there are two different communications between a NIC and a driver of a host: one at the receiving end and the other at the sending end. It is during these communications that the pinned down buffers are to be released at each end after the message data has been successfully transferred (Please see Fig. 7 in Pandya and the examiner's response to rejection of claim 20 in Ground V). Just as Osborne's "Send" command includes "source address" as STag (pointer) to message buffers on the source side, likewise, Pandya's "Send completion status and sense" from receiving host to receiving NIC will include pointer to received message buffers.

In response to appellants' allegation on page 44 of the appeal brief, the examiner would like to refer to Figs. 8-9 of Roach et al. reference that show use of pointers (Buffer Starting Word Address [BSWA]) for the allocated buffers, which is what the appellants' define as STag. Furthermore, Roach et al reference is not wholly unrelated to identifying fields in FC2 (Fiber Channel 2) header, as can be seen in Fig. 7 that includes current buffer list address and different control parameters in the FC2 header.

Accordingly, independent claim 25 is obvious over Osborne in view of Pandya and Roach, and therefore not allowable.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/K. G. B./
Examiner, Art Unit 2443

December 2, 2010

Conferees:

/J Bret Dennison/
Primary Examiner, Art Unit 2443

/Tonia LM Dollinger/
Supervisory Patent Examiner, Art Unit 2443